A Difference of Convex Functions Algorithm for Switched Linear Regression

Tao Pham Dinh, Hoai Minh Le, Hoai An Le Thi, Fabien Lauer

▶ **To cite this version:**

Tao Pham Dinh, Hoai Minh Le, Hoai An Le Thi, Fabien Lauer. A Difference of Convex Functions Algorithm for Switched Linear Regression. IEEE Transactions on Automatic Control, 2014, 10.1109/TAC.2014.2301575 . hal-00931206

HAL Id: hal-00931206
https://hal.science/hal-00931206

Submitted on 15 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Difference of Convex Functions Algorithm for Switched Linear Regression

Tao PHAM DINH, Hoai Minh LE, Hoai An LE THI, and Fabien LAUER

***Abstract*—This paper deals with switched linear system identification and more particularly aims at solving switched linear regression problems in a large-scale setting with both numerous data and many parameters to learn. We consider the recent minimum-of-error framework with a quadratic loss function, in which an objective function based on a sum of minimum errors with respect to multiple submodels is to be minimized. The paper proposes a new approach to the optimization of this nonsmooth and nonconvex objective function, which relies on Difference of Convex (DC) functions programming. In particular, we formulate a proper DC decomposition of the objective function, which allows us to derive a computationally efficient DC algorithm. Numerical experiments show that the method can efficiently and accurately learn switching models in large dimensions and from many data points.**

***Index Terms*—Switched linear systems, Piecewise affine systems, System identification, Switched regression, Nonconvex optimization, Nonsmooth optimization, DC programming, DCA.**

I. Introduction

We consider switched regression as the problem of learning a collection of n models from a training set of N data pairs, $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$, generated by a switching function as

$$y_i = f_{\lambda_i}(\boldsymbol{x}_i) + e_i, \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^p$ is the regression vector, $y_i \in \mathbb{R}$ is the observed output, $\lambda_i \in \{1, \ldots, n\}$ is the *mode* determining which one of the n functions $\{f_j\}_{j=1}^n$ was active when computing the output y_i for the ith data point and e_i is an additive noise term. In particular, the aim of the paper is to estimate the parameters of the submodels $\{f_j\}_{j=1}^n$ from such a data set, under the assumptions that the parametric form of the submodels are known but that the mode λ_i of each data pair $(\boldsymbol{x}_i, y_i)$ is unknown. In the context of system identification, we focus on a class of hybrid systems known as multiple-input-single-output (MISO) arbitrarily Switched AutoRegressive with eXogenous input (SARX) systems of orders n_a and n_b. In this case, the regression vector is built from past inputs $\boldsymbol{u}_{i-k} \in \mathbb{R}^{n_u}$ and outputs y_{i-k}, i.e., $\boldsymbol{x}_i = [y_{i-1} \ \ldots \ y_{i-n_a}, \ \boldsymbol{u}_i^T \ \ldots \ \boldsymbol{u}_{i-n_b}^T]^T$ and can be in high dimension depending on the number of inputs n_u. We further assume that the number of modes n is *a priori* fixed. Note that, even with a known n and linear submodels f_j, the problem remains complex and amounts to solving a nonconvex optimization program. This difficulty is due to the intrinsic combination of two subproblems: the unsupervised classification of the points into modes and the estimation of a submodel for each mode.

Related work. Over the last decade, the control community showed an increasing interest in switched regression as a means to identify hybrid (or switched) dynamical systems. These dynamical systems are described by a collection of subsystems and a switching mechanism resulting in a convenient framework for the study of the complex nonlinear behaviors of cyber-physical systems. However, their identification, i.e., the estimation of their parameters from experimental data, remains in most cases an open issue, which can, for SARX systems and through an appropriate choice of regressors in $\boldsymbol{x}_i$, be posed as the problem of learning the n functions f_j in the model (1). Therefore, most methods for switched regression were proposed in this context for hybrid systems with linear subsystems [1], [2], [3], [4], [5]. More recently, the continuous optimization framework of [6] offered a convenient approach for problems with large data sets and was extended in [7] to deal with nonlinear subsystems. Beside [6], the current trend, see, e.g., [8], [9], [10], [11], [12], seems to focus on convex formulations in order to avoid local minima issues. However, these approaches are often based on relaxations of nonconvex optimization problems which typically depend on specific conditions on the data in order to guarantee the equivalence with the original problem; and these conditions can be difficult to verify or obtain in practice. Moreover, in spite of this activity, the issue of learning large-scale models with numerous modes and/or with a high-dimensional regression vector remains largely unanswered. For instance, the sparse optimization-based method of [8] relies on a condition on the fraction of data generated by each mode, which is not easily satisfied when the number of modes becomes large. Regarding the nonconvex optimization-based methods, including the one of [6], their computational burden quickly becomes prohibitive when the number of parameters to estimate becomes large.

Paper contribution. This paper proposes an algorithm dedicated to the minimization of a regularized version of the cost function considered in [6], which is both nonsmooth and nonconvex. In [6], the estimation is originally performed through the use of a generic global optimization algorithm – the Multilevel Coordinate Search (MCS) [13]. Though rather effective for small-size problems with few parameters to learn, this approach is not applicable to large-scale systems. On the contrary, our approach is devised explicitly to deal with such cases. The proposed method is based on DC (Difference of Convex functions) programming and DCA (DC Algorithms) that were introduced by Pham Dinh Tao in their preliminary form in 1985. They have been extensively developed since

T. Pham Dinh is with the Laboratory of Modelling, Optimization & Operations Research National Institute for Applied Sciences - Rouen, Avenue de l'Université- 76801 Saint-Etienne-du-Rouvray cedex, France

H.M. Le and H.A. Le Thi are with the Laboratory of Theoretical and Applied Computer Science - LITA EA 3097 University of Lorraine, Ile de Saulcy, 57045 Metz, France

F. Lauer is with the LORIA, Université de Lorraine, CNRS, Inria, F-54506 Vandœuvre-lès-Nancy, France

1994 by Le Thi Hoai An and Pham Dinh Tao and become now classic and increasingly popular (see e.g. [14], [15], [16], [17], [18]). Our motivation is based on the fact that DCA is a fast and scalable approach which has been successfully applied to many large-scale (smooth or nonsmooth) nonconvex programs in various domains of applied sciences, in particular in data analysis and data mining, for which it provided quite often a global solution and proved to be more robust and efficient than standard methods (see [15], [16], [17], [14], [18] and references therein). Using a natural DC decomposition of the cost function, we devise an efficient and inexpensive DC algorithm that can solve large scale problems.

Paper organization. The paper starts by introducing the considered framework for switched regression and the main optimization problem in Sect. II. Then, we formulate the proposed DC programming approach and DC algorithm in Sect. III. The paper ends with numerical experiments in Sect. IV and conclusions in Sect. V.

II. Switched regression framework

Under the assumption that the number of modes n is known, the switched regression problem is to find a collection of n models $\{f_j\}_{j=1}^n$ that best fits the given collection of data points $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \in (\mathbb{R}^p \times \mathbb{R})^N$. In order to control the complexity of the models f_j in large dimensions p, we additionally consider a regularized version of this problem. This can be posed as a mixed-integer nonlinear programming problem of the form

$$\begin{aligned} \min_{\{f_j\},\{\beta_{ij}\}} \ & \sum_{i=1}^{N}\sum_{j=1}^{n} \beta_{ij}\ell(y_i - f_j(\boldsymbol{x}_i)) \ + \ \gamma \sum_{j=1}^{n} \mathcal{R}(f_j) \qquad (2) \\ \text{s.t.} \ \ & \beta_{ij} \in \{0,1\}, \ i = 1,\ldots,N, \ j = 1,\ldots,n, \\ & \sum_{j=1}^{n} \beta_{ij} = 1, \ i = 1,\ldots,N, \end{aligned}$$

where β_{ij} is a binary variable coding the assignment of the point of index i to mode j, ℓ is a suitable loss function and $\mathcal{R}(f_j)$ is a convex regularization term, weighted by a trade-off parameter $\gamma \geq 0$. From the solution to (2), the mode of each data point is recovered via the binary variables by $\lambda_i = \arg\max_{j=1,\ldots,n} \beta_{ij}$.

As proposed by [6], this problem can be reformulated to give rise to the Minimum-of-Error (ME) estimator, defined as the solution to[1]

$$\min_{\{f_j\}} \sum_{i=1}^{N} \min_{j\in\{1,\ldots,n\}} \ell(y_i - f_j(\boldsymbol{x}_i)) \ + \ \gamma \sum_{j=1}^{n} \mathcal{R}(f_j). \qquad (3)$$

This formulation explicitly includes the solution of the classification subproblem with respect to the β_{ij} as

$$\begin{aligned} \forall i \in \{1,\ldots,N\}, \quad & \beta_{i\hat{\lambda}_i} = 1, \\ \forall i \in \{1,\ldots,N\}, \forall j \in \{1,\ldots,n\} \setminus \hat{\lambda}_i, \quad & \beta_{ij} = 0, \end{aligned}$$

[1]In [6], the focus is on small dimensions p and the ME estimator is defined without the regularization terms (with $\gamma = 0$).

where

$$\forall i \in \{1,\ldots,N\}, \ \hat{\lambda}_i = \arg \min_{j\in\{1,\ldots,n\}} \ell(y_i - f_j(\boldsymbol{x}_i)). \qquad (4)$$

The classification rule (4) states that a data point $\boldsymbol{x}_i$ must be associated to the mode j for which the corresponding submodel f_j yields the best estimate of the target output y_i.

Compared with (2), the ME estimator (3) significantly reduces the influence of the number of data N on the complexity of the problem and the time required to find its solution. In particular, the number of variables does not depend on N and no binary variables are involved.

In the remaining of the paper, we focus on linear submodels

$$f_j(\boldsymbol{x}_i) = \boldsymbol{w}_j^T \boldsymbol{x}_i, \qquad (5)$$

with parameter vectors $\boldsymbol{w}_j \in \mathbb{R}^p$ and a regularization based on the ℓ_2-norm of these vectors, i.e., $\mathcal{R}(f_j) = \|\boldsymbol{w}_j\|_2^2$. This regularization term is classicaly used in ridge regression and is particularly useful when estimating models in large dimensions from small data sets. We further concentrate on the quadratic loss function, $\ell(e) = e^2$. Thus, we aim at solving

$$\min_{\boldsymbol{w}\in\mathbb{R}^{np}} \ J(\boldsymbol{w}) := \sum_{i=1}^{N} \min_{j\in\{1,\ldots,n\}} (y_i - \boldsymbol{w}_j^T \boldsymbol{x}_i)^2 + \gamma \|\boldsymbol{w}\|_2^2, \ (6)$$

where the vector of variables $\boldsymbol{w} = [\boldsymbol{w}_1^T, \ldots, \boldsymbol{w}_n^T]^T$ is of dimension np and contains all the parameter vectors $\boldsymbol{w}_j$ to be estimated, which yields $\|\boldsymbol{w}\|_2^2 = \sum_{j=1}^n \|\boldsymbol{w}_j\|_2^2$.

III. DC programming approach

In this section, we give a brief introduction to DC programming and DCA for an easy understanding of these tools and our motivation to use them for solving Problem (6).

A. A brief introduction to DC programming and DCA

DC Programming and DCA constitute the backbone of smooth/nonsmooth nonconvex programming and global optimization. They address DC programs of the form

$$\alpha = \inf\{f(\boldsymbol{w}) := g(\boldsymbol{w}) - h(\boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^d\}, \quad (P_{dc}) \qquad (7)$$

where g and h are lower semicontinuous proper convex functions on $\mathbb{R}^d$. Such a function f is called a DC function and $g-h$ a DC decomposition of f, while g and h are DC components of f. Recall the natural convention $+\infty - (+\infty) = +\infty$ in DC programming, and that a DC program with a closed convex constraint set $C \subset \mathbb{R}^d$,

$$\beta = \inf\{\varphi(\boldsymbol{w}) - \phi(\boldsymbol{w}) : \boldsymbol{w} \in C\},$$

can be rewritten in the form of (P_{dc}) as

$$\beta = \inf\{g(\boldsymbol{w}) - h(\boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^d\},$$

where $g := \varphi + \chi_C$, $h := \phi$ and χ_C stands for the indicator function of C, i.e., $\chi_C(\boldsymbol{u}) = 0$ if $\boldsymbol{u} \in C$, and $+\infty$ otherwise. Let

$$g^*(\boldsymbol{v}) := \sup\{\langle \boldsymbol{w}, \boldsymbol{v}\rangle - g(\boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^d\}$$

be the Fenchel conjugate function of g. Then, the following program is called the dual program of (P_{dc}):

$$\alpha_D = \inf\{h^*(\boldsymbol{v}) - g^*(\boldsymbol{v}) : \boldsymbol{v} \in \mathbb{R}^d\}. \quad (D_{dc}) \tag{8}$$

One can prove (see, e.g., [16]) that $\alpha = \alpha_D$ and that there is a perfect symmetry between primal and dual DC programs: the dual to (D_{dc}) is exactly (P_{dc}).

For a convex function θ, the subdifferential of θ at $\boldsymbol{w}_0 \in \text{dom}\ \theta := \{\boldsymbol{w} \in \mathbb{R}^d : \theta(\boldsymbol{w}_0) < +\infty\}$, denoted by $\partial\theta(\boldsymbol{w}_0)$, is defined by

$$\partial\theta(\boldsymbol{w}_0) := \{\boldsymbol{v} \in \mathbb{R}^d : \theta(\boldsymbol{w}) \geq \theta(\boldsymbol{w}_0) + \langle \boldsymbol{w} - \boldsymbol{w}_0, \boldsymbol{v}\rangle, \forall \boldsymbol{w} \in \mathbb{R}^d\}.$$

The subdifferential $\partial\theta(\boldsymbol{w}_0)$ generalizes the derivative in the sense that θ is differentiable at $\boldsymbol{w}_0$ if and only if $\partial\theta(\boldsymbol{w}_0) \equiv \{\bigtriangledown_{\boldsymbol{w}}\theta(\boldsymbol{w}_0)\}$. Recall the well-known property [16] related to subdifferential calculus of a convex function θ:

$$\boldsymbol{v}_0 \in \partial\theta(\boldsymbol{w}_0) \Leftrightarrow \boldsymbol{w}_0 \in \partial\theta^*(\boldsymbol{v}_0) \Leftrightarrow \langle \boldsymbol{w}_0, \boldsymbol{v}_0\rangle = \theta(\boldsymbol{w}_0) + \theta^*(\boldsymbol{v}_0). \tag{9}$$

The complexity of DC programs resides, of course, in the lack of practical global optimality conditions. Local optimality conditions are then useful in DC programming.

A point $\boldsymbol{w}^*$ is said to be *a local minimizer* of $g - h$ if $g(\boldsymbol{w}^*) - h(\boldsymbol{w}^*)$ is finite and there exists a neighbourhood $\mathcal{U}$ of $\boldsymbol{w}^*$ such that

$$g(\boldsymbol{w}^*) - h(\boldsymbol{w}^*) \leq g(\boldsymbol{w}) - h(\boldsymbol{w}), \quad \forall \boldsymbol{w} \in \mathcal{U}.$$

The necessary local optimality condition for (primal) DC program (P_{dc}) is given by

$$\emptyset \neq \partial h(\boldsymbol{w}^*) \subset \partial g(\boldsymbol{w}^*). \tag{10}$$

The condition (10) is also sufficient (for local optimality) in many important classes of DC programs (see [15], [14]).

A point $\boldsymbol{w}^*$ is said to be *a critical point* of $g - h$ if

$$\partial h(\boldsymbol{w}^*) \cap \partial g(\boldsymbol{w}^*) \neq \emptyset. \tag{11}$$

The relation (11) is in fact the generalized KKT condition for (P_{dc}) and w^* is also called a generalized KKT point.

Philosophy of DCA: Based on local optimality conditions and duality in DC programming, the DCA consists in constructing two sequences $\{\boldsymbol{w}^l\}$ and $\{\boldsymbol{v}^l\}$ of trial solutions to the primal and dual programs respectively, such that the sequences $\{g(\boldsymbol{w}^l) - h(\boldsymbol{w}^l)\}$ and $\{h^*(\boldsymbol{v}^l) - g^*(\boldsymbol{v}^l)\}$ are decreasing, and $\{\boldsymbol{w}^l\}$ (resp. $\{\boldsymbol{v}^l\}$) converges to a primal feasible solution $\boldsymbol{w}^*$ (resp. a dual feasible solution $\boldsymbol{v}^*$) satisfying local optimality conditions and

$$\boldsymbol{w}^* \in \partial g^*(\boldsymbol{v}^*), \quad \boldsymbol{v}^* \in \partial h(\boldsymbol{w}^*). \tag{12}$$

Thus, according to (9) and (12), $\boldsymbol{w}^*$ and $\boldsymbol{v}^*$ are critical points of $g - h$ and $h^* - g^*$, respectively.

The main idea behind DCA is to replace in the primal DC program (P_{dc}), at the current point $\boldsymbol{w}^l$ of iteration l, the second component h with its affine minorization defined by

$$h_l(\boldsymbol{w}) := h(\boldsymbol{w}^l) + \langle \boldsymbol{w} - \boldsymbol{w}^l, \boldsymbol{v}^l\rangle, \quad \boldsymbol{v}^l \in \partial h(\boldsymbol{w}^l)$$

to give rise to the primal convex program of the form

$$\begin{aligned}(P_l) \quad & \inf\{g(\boldsymbol{w}) - h_l(\boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^d\} \\ \Leftrightarrow\ & \inf\{g(\boldsymbol{w}) - \langle \boldsymbol{w}, \boldsymbol{v}^l\rangle : \boldsymbol{w} \in \mathbb{R}^d\},\end{aligned}$$

an optimal solution of which is taken as $\boldsymbol{w}^{l+1}$.

Dually, a solution $\boldsymbol{w}^{l+1}$ of (P_l) is then used to define the dual convex program (D_{l+1}) obtained from (D_{dc}) by replacing g^* with its affine minorization defined by

$$(g^*)_l(\boldsymbol{v}) := g^*(v^l) + \langle \boldsymbol{v} - \boldsymbol{v}^l, \boldsymbol{w}^{l+1}\rangle, \quad \boldsymbol{w}^{l+1} \in \partial g^*(\boldsymbol{v}^l)$$

to obtain the convex program

$$(D_{l+1}) \quad \inf\{h^*(\boldsymbol{v}) - [g^*(\boldsymbol{v}^l) + \langle \boldsymbol{v} - \boldsymbol{v}^l, \boldsymbol{w}^{l+1}\rangle] : \boldsymbol{v} \in \mathbb{R}^d\}$$

an optimal solution of which is taken as $\boldsymbol{v}^{l+1}$. The process is repeated until convergence.

Overall, DCA performs a double linearization with the help of the subgradients of h and g^*. According to relation (9) it is easy to see that the optimal solution set of (P_l) (resp. (D_{l+1})) is nothing but $\partial g^*(\boldsymbol{v}^l)$ (resp. $\partial h(\boldsymbol{w}^{l+1})$). Hence, we can say that DCA is an iterative primal-dual subgradient method that yields the next scheme: (starting from given $\boldsymbol{w}^0 \in \text{dom}\ \partial h$)

$$\boldsymbol{v}^l \in \partial h(\boldsymbol{w}^l); \quad \boldsymbol{w}^{l+1} \in \partial g^*(\boldsymbol{v}^l),\ \forall l \geq 0. \tag{13}$$

A deeper insight into DCA has been described in [14]. The generic DCA scheme is shown below.

Algorithm 1 DCA

Initialization: Let $\boldsymbol{w}^0 \in \mathbb{R}^d$ be an initial vector (possibly drawn randomly), $l \leftarrow 0$.
repeat
Calculate $\boldsymbol{v}^l \in \partial h(\boldsymbol{w}^l)$.
Calculate

$$\boldsymbol{w}^{l+1} \in \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} g(\boldsymbol{w}) - h(\boldsymbol{w}^l) - \langle \boldsymbol{w} - \boldsymbol{w}^l, \boldsymbol{v}^l\rangle \quad (P_l)$$

$l \leftarrow l + 1$.
until convergence of $\boldsymbol{w}^l$.

Convergence properties of DCA and its theoretical basis can be found in [14], [15], [16], [17], [18]. For instance, it is important to mention the following properties:

(i) DCA is a descent method (the sequences $\{g(\boldsymbol{w}^l) - h(\boldsymbol{w}^l)\}$ and $\{h^*(\boldsymbol{v}^l) - g^*(\boldsymbol{v}^l)\}$ are decreasing) without linesearch;
(ii) if the optimal value α of the problem (P_{dc}) is finite and the infinite sequences $\{\boldsymbol{w}^l\}$ and $\{\boldsymbol{v}^l\}$ are bounded, then every limit point $\boldsymbol{w}^*$ (resp. $\boldsymbol{v}^*$) of the sequence $\{\boldsymbol{w}^l\}$ (resp. $\{\boldsymbol{v}^l\}$) is a critical point of $g - h$ (resp. $h^* - g^*$);
(iii) DCA has a linear convergence for general DC programs and has a finite convergence for polyhedral DC programs.

DCA's distinctive feature relies upon the fact that DCA deals with the convex DC components g and h but not with the DC function f itself. Moreover, a DC function f *has infinitely many DC decompositions (and there are as many DCA as there are equivalent DC programs and their DC decompositions) which have crucial implications for the qualities* (speed of convergence, robustness, efficiency, globality of computed solutions,...) *of DCA*. Finding an appropriate equivalent DC program and a suitable DC decomposition is consequently important from the algorithmic point of view. For a complete

study of DC programming and DCA the reader is referred to [14], [15], [16], [17], [18] and references therein.

The solution of a nonconvex program by DCA must be composed of two stages: the search of both a suitable DC program and its relevant DC decomposition, and the choice of a strategy for a good initial point, taking into account the specific structure of the nonconvex program. In this paper, by exploiting a well-crafted DC decomposition for problem (6), we design a computationally inexpensive DCA scheme: each iteration requires only to solve an unconstrained and strongly convex quadratic program which is separable in the components $\boldsymbol{w}_j$ of $\boldsymbol{w}$.

B. A DC formulation of Problem (6)

Let us denote

$$J_i(\boldsymbol{w}) = \min_{j\in\{1,\ldots,n\}} (y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2.$$

We can write the function $J_i(\boldsymbol{w})$ in the form

$$J_i(\boldsymbol{w}) = \sum_{j=1}^{n}(y_i-\boldsymbol{w}_j^T\boldsymbol{x}_i)^2 - \max_{j\in\{1,\ldots,n\}} \sum_{k\in\{1,\ldots,n\}\setminus j} (y_i-\boldsymbol{w}_k^T\boldsymbol{x}_i)^2.$$

Consequently, the objective function of (6) can be written as

$$J(\boldsymbol{w}) = G(\boldsymbol{w}) - H(\boldsymbol{w}),$$

where

$$G(\boldsymbol{w}) = \sum_{i=1}^{N}\sum_{j=1}^{n}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2 + \gamma\sum_{j=1}^{n}\|\boldsymbol{w}_j\|_2^2$$

and

$$H(\boldsymbol{w}) = \sum_{i=1}^{N}\max_{j\in\{1,\ldots,n\}} \sum_{k\in\{1,\ldots,n\}\setminus j} (y_i - \boldsymbol{w}_k^T\boldsymbol{x}_i)^2$$

are convex functions. Hence, we can recast Problem (6) as the following DC program

$$\min\left\{G(\boldsymbol{w}) - H(\boldsymbol{w}) : \boldsymbol{w}\in\mathbb{R}^{np}\right\}. \tag{14}$$

C. A DCA scheme

According to Section III-A, applying DCA to (14) amounts to computing the two sequences $\left\{\boldsymbol{w}^l\right\}$ and $\left\{\boldsymbol{v}^l\right\}$ such that

$$\boldsymbol{v}^l \in \partial H(\boldsymbol{w}^l), \tag{15}$$

$$\boldsymbol{w}^{l+1} \in \arg\min_{\boldsymbol{w}\in\mathbb{R}^{np}} G(\boldsymbol{w}) - \langle\boldsymbol{v}^l, \boldsymbol{w}\rangle. \tag{16}$$

Problem (16) is a convex quadratic program whose optimal solution can be determined in an inexpensive way. Indeed, by defining the target output vector $\boldsymbol{y} = [y_1,\ldots,y_N]^T$ and the regression matrix $\boldsymbol{X} = [\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N]^T$, we have

$$\begin{aligned} G(\boldsymbol{w}) &= \sum_{i=1}^{N}\sum_{j=1}^{n}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2 + \gamma\sum_{j=1}^{n}\|\boldsymbol{w}_j\|_2^2 \\ &= \sum_{j=1}^{n}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}_j)^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}_j) + \gamma\boldsymbol{w}_j^T\boldsymbol{w}_j \\ &= G_0 + \sum_{j=1}^{n} G_j(\boldsymbol{w}_j). \end{aligned}$$

where $G_0 = n\boldsymbol{y}^T\boldsymbol{y}$ is a constant and $G_j(\boldsymbol{w}_j) = \boldsymbol{w}_j^T(\boldsymbol{X}^T\boldsymbol{X} + \gamma\boldsymbol{I})\boldsymbol{w}_j - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w}_j$ is a function of the subset of variables $\boldsymbol{w}_j$.

Hence, the objective function of problem (16) is separable with respect to n groups of variables $\{\boldsymbol{w}_j\}_{j=1}^n$ and solving (16) amounts to solving n separate optimization problems. More precisely, for $j = 1,\ldots,n$, $\boldsymbol{w}_j^{l+1}$ is the solution to the unconstrained convex quadratic program

$$\min_{\boldsymbol{w}_j\in\mathbb{R}^p} \boldsymbol{w}_j^T(\boldsymbol{X}^T\boldsymbol{X} + \gamma\boldsymbol{I})\boldsymbol{w}_j - (2\boldsymbol{y}^T\boldsymbol{X} + \boldsymbol{v}_j^T)\boldsymbol{w}_j,$$

where $\boldsymbol{I}$ stands for the identity matrix of appropriate size.

For (15), we compute a subgradient $\boldsymbol{v}\in\partial H(\boldsymbol{w})$ as follows:

$$\boldsymbol{v}\in\partial H(\boldsymbol{w}) \Leftrightarrow \boldsymbol{v}\in\sum_{i=1}^{N}\partial h_i(\boldsymbol{w}),$$

where $h_i(\boldsymbol{w}) = \max_{j\in\{1,\ldots,n\}} h_i^j(\boldsymbol{w})$ with $h_i^j(\boldsymbol{w}) = \sum_{k\in\{1,\ldots,n\}\setminus j}(y_i - \boldsymbol{w}_k^T\boldsymbol{x}_i)^2$.

Let $\mathcal{J}_i(\boldsymbol{w}) = \{j\in\{1,\ldots,n\} : h_i^j(\boldsymbol{w}) = h_i(\boldsymbol{w})\}$. We have

$$\partial h_i(\boldsymbol{w}) = co\left\{\bigcup_{j\in\mathcal{J}_i(\boldsymbol{w})}\partial h_i^j(\boldsymbol{w})\right\},$$

where co stands for the convex hull. Hence, $\partial h_i(\boldsymbol{w})$ is a convex combination of $\{\nabla h_i^j(\boldsymbol{w}) : j\in\mathcal{J}_i(\boldsymbol{w})\}$, i.e.,

$$\partial h_i(\boldsymbol{w}) = \sum_{j\in\mathcal{J}_i(\boldsymbol{w})}\mu_i^j\nabla h_i^j(\boldsymbol{w}); \quad \sum_{j\in\mathcal{J}_i(\boldsymbol{w})}\mu_i^j = 1,\ \mu_i^j\geq 0.$$

In particular, in our implementation, a subgradient $\boldsymbol{\eta}_i$ of $h_i(\boldsymbol{w})$ is chosen as follows:

$$\boldsymbol{\eta}_i = \nabla h_i^{j_0}(\boldsymbol{w}),\ \text{with } j_0\in\mathcal{J}_i(\boldsymbol{w}).$$

From the above computations, the DCA applied to problem (14) is described via Algorithm 2, where the set valued function $\mathcal{J}_i$ is never computed as a whole but only evaluated at a given $\boldsymbol{w}^l$ for each iteartion l.

Algorithm 2 ME-DCA

Initialization: Draw a random $\boldsymbol{w}^0\in\mathbb{R}^{np}$. Let $\tau > 0$ be sufficiently small. $l\leftarrow 0$.

repeat

Set $\boldsymbol{v}^l = \sum_{i=1}^{N}\nabla h_i^{j_0}(\boldsymbol{w}^l)$, with $j_0\in\mathcal{J}_i(\boldsymbol{w}^l)$.

Compute $\boldsymbol{w}^{l+1}$ by solving the n unconstrained convex quadratic programs, i.e., for $j = 1,\ldots,n$:

$$\boldsymbol{w}_j^{l+1}\in\arg\min_{\boldsymbol{w}_j\in\mathbb{R}^p}\boldsymbol{w}_j^T(\boldsymbol{X}^T\boldsymbol{X}+\gamma\boldsymbol{I})\boldsymbol{w}_j - (2\boldsymbol{y}^T\boldsymbol{X}+\boldsymbol{v}_j^l)\boldsymbol{w}_j.$$

Increase the iteration counter: $l\leftarrow l+1$.

until $\|\boldsymbol{w}^{l+1} - \boldsymbol{w}^l\|/(\|\boldsymbol{w}^l\|+1)\leq\tau$.

return $J(\boldsymbol{w}^l)$ and $\boldsymbol{w}^l$.

Convergence and complexity of the algorithm: From general convergence properties of DCA, the ME-DCA has a linear convergence. One of the key points of DCA is that it does not rely on a line search strategy. Therefore, there is no need to evaluate the objective function numerous times as in standard gradient descent schemes, for instance. One iteration of the ME-DCA algorithm only relies on few basic operations, which leads to a low computational cost. Indeed, at each iteration the computation of $\partial H(\boldsymbol{w}^l)$ requires $N(np+(n-1)p^2)$ operations, and the solution of n separate unconstrained convex quadratic programs of size p, which is equivalent to solve n linear systems of size p, has a total computing cost $\mathcal{O}(np^2)$. Therefore the complexity of ME-DCA is $\mathcal{O}\left(N(np+(n-1)p^2)+np^2\right)=\mathcal{O}\left(Nnp^2\right)$.

IV. Numerical experiments

In this Section, we compare the ME-DCA algorithm with ME-MCS, i.e., the optimization of (6) by the MCS algorithm [13] as proposed by [6] with default parameters.[2] The proposed ME-DCA is initialized with $\boldsymbol{w}_j^0 = [w_{j1}^0,\ldots,w_{jp}^0]$ randomly drawn from a uniform distribution with $\min_{i\in\{1,\ldots,N\}} x_{ik} \leq w_{jk}^0 \leq \max_{i\in\{1,\ldots,N\}} x_{ik}$, $k=1,\ldots,p$. DCA is stopped with the tolerance $\tau = 10^{-6}$. Since all the tests below consider a large-scale setting with a sufficiently large number of data points with respect to the number of parameters, both the ME-DCA and ME-MCS use an unregularized version of the method (i.e., with $\gamma=0$).

The data are generated by $y_i = \boldsymbol{\theta}_{\lambda_i}^T \boldsymbol{x}_i + e_i$, $i=1,\ldots,N$, where the $\boldsymbol{\theta}_j \in \mathbb{R}^p$, $j=1,\ldots,n$, are the true parameters to recover, λ_i is the true mode of point i uniformly distributed in $\{1,\ldots,n\}$, and $e_i \sim \mathcal{N}(0,\sigma_e^2)$ is a Gaussian noise with variance $\sigma_e^2 = 0.1$ (corresponding to a signal-to-noise ratio of about 27 dB). The methods are compared on the basis of the computing time and the normalized mean squared error on the parameters, NMSE $= \sum_{j=1}^n \|\boldsymbol{\theta}_j - \boldsymbol{w}_j\|_2^2 / \|\boldsymbol{\theta}_j\|_2^2$. In the Tables, we report the mean and the standard deviation of the NMSE over 100 experiments with different sets of true parameters $\{\boldsymbol{\theta}_j\}$ and noise sequences. Note that since the goal is to find a global solution to a nonconvex optimization problem, we cannot guarantee the success of the method, which in some cases may yield a local and unsatisfactory solution. Therefore, we also measure the performance of the algorithms through their ability to obtain a satisfactory solution that is not too far from the global one. In particular, the percentage of success over the multiple experiments is estimated by the percentage of experiments for which the mean squared error, MSE $= 1/N \sum_{i=1}^N (y_i - \boldsymbol{w}_{\hat{\lambda}_i}^T \boldsymbol{x}_i)^2$, where the $\hat{\lambda}_i$ are estimated by (4) and the $\boldsymbol{w}_j$ are the learned parameters, satisfies MSE $< 2\times$ MSEref, where MSEref stands for the MSE of the reference model trained by applying n independent least squares estimators to the data grouped in n subsets on the basis of the true classification. Note that, since the precise modeling error is irrelevant for unsuccessful cases, the average NMSE is computed from the successful cases only. All computing times refer to Matlab implementations of the methods running on a standard desktop computer.

[2]Software available at http://www.loria.fr/~lauer/software.html .

TABLE I
Average NMSE and computing time over 100 experiments with large data sets of size N ($n=3$, $p=4$).

	ME-DCA			ME-MCS		
N	NMSE ($\times 10^{-6}$)	Succ. (%)	Time (sec.)	NMSE ($\times 10^{-6}$)	Success (%)	Time (sec.)
100	12±4	80	**0.1**	**10±5**	85	4.5
1000	**7±3**	96	**1.1**	10±2	92	3.6
5000	**0.1±0.05**	**100**	10	**0.1±0.05**	**100**	**8**
10000	3±1.5	**99**	**30**	4±2	95	35
50000	**0.1±0.05**	**100**	**55**	**0.1±0.05**	95	65

A. Large data sets

We start by comparing the methods with respect to their ability to deal with large data sets. In particular, the number of data N is increased from 100 to 50 000. The results are summarized in Table I, where the NMSE, the percentage of successful experiments and the computing time are reported. These results show that, by sharing the same problem formulation (6), ME-DCA benefits from the ME-MCS ability to deal efficiently with large data sets, with however a slightly lower computational cost. In addition to this increase of efficiency, ME-DCA is at least as accurate as ME-MCS in terms of both the percentage of successful trainings and the model error.

B. Large models

The computing time of previous methods such as ME-MCS heavily relies on the number of model parameters $n \times p$. Thus, these methods may not be suitable for large models with numerous modes or regressors. In the following experiments, the dimension of the data p and the number of modes n are both increased to test the ability of the proposed ME-DCA to efficiently and accurately learn large models. Table II shows results for models with up to 2 000 parameters trained on 10 000 data points. These results clearly indicate that ME-DCA can tackle problems with much larger dimensions n and p than the classical ME-MCS algorithm, which does not yield a solution after 2 hours in many cases. For moderate dimensions, such as $n=5$ and $5 \leq p \leq 20$, ME-DCA is also much faster than ME-MCS (between 15 and 100 times faster) without being less accurate. Indeed, the percentage of success is always higher for ME-DCA than for ME-MCS.

C. Switched system identification

We now turn to switched dynamical system identification. In this case, the regression vectors are given by $\boldsymbol{x}_i = [y_{i-1} \ \ldots \ y_{i-n_a} \ \boldsymbol{u}_i^T \ \ldots \ \boldsymbol{u}_{i-n_b}^T]^T$ and are constrained to lie on a particular manifold, which could affect the results of DCA. However, as shown by Table III, ME-DCA provides accurate system identification results that are again better than those obtained with the original ME-MCS algorithm. These results are obtained with 10 000 data points generated by second-order systems ($n_a = n_b = 2$) with various numbers of modes and of inputs, and with random parameters[3] uniformly

[3]Sets of parameters generating diverging trajectories are discarded.

TABLE II
AVERAGE NMSE, PERCENTAGE OF SUCCESS AND COMPUTING TIME OVER 100 EXPERIMENTS WITH LARGE MODELS.

		ME-DCA			ME-MCS		
n	p	NMSE ($\times 10^{-6}$)	Succ. (%)	Time (sec.)	NMSE ($\times 10^{-6}$)	Succ. (%)	Time (sec.)
3	100	**0.1±0.1**	**100**	**31**	N/A	0	7200
3	200	**0.1±0.1**	**100**	**112**	N/A	0	7200
5	5	10±2	95	**4**	14±2	89	219
5	10	5±2	88	**18**	7±1	65	290
5	20	3±1	85	**3**	5±1	75	270
5	50	**0.1±0.1**	**100**	**180**	N/A	0	7200
10	200	**12±2**	**84**	**87**	N/A	0	7200
20	100	**9±3**	**92**	**440**	N/A	0	7200

TABLE III
AVERAGE NMSE AND PERCENTAGE OF SUCCESS OVER 100 EXPERIMENTS FOR VARYING NUMBER OF MODES (n) AND NUMBER OF INPUTS (n_u).

		ME-DCA		ME-MCS	
n	n_u	NMSE ($\times 10^{-6}$)	Succ. (%)	NMSE ($\times 10^{-6}$)	Succ. (%)
3	10	2 ± 1	**80**	3 ± 1	75
3	30	7 ± 2	**74**	10 ± 3	68
5	3	3 ± 1	**82**	4 ± 2	**82**
10	20	14 ± 3	**65**	N/A	0

distributed in $[-1, 1]$. With $n = 5$ and $n_u = 10$, Figure 1 also shows that, for reasonable values of the signal-to-noise ratio above 12 dB, the noise level has little influence on the quality of the ME-DCA solution: the success rate remains above 73% and higher than the one of ME-MCS. In addition, in these experiments, the solution is always obtained in seconds with ME-DCA instead of minutes or hours with ME-MCS.

V. CONCLUSION

We proposed a new optimization algorithm for switched linear regression in the minimum-of-error framework. The proposed DC algorithm efficiently deals with both the nonconvexity and the nonsmoothness of the objective function. Compared with previous approaches, the algorithm is particularly efficient for learning large models with many modes and/or parameters.

However, only the convergence towards a local minimum can be guaranteed. Though promising results were obtained on multiple examples with high success rates, the probability of success could be further analyzed as in [19] and compared with the one of the k-LinReg algorithm proposed in that paper. In addition, a method that is guaranteed to find the global solution for small dimensions would be of primary interest and is the subject of ongoing investigations on branch-and-bound DC programming. In comparison with the mixed-integer programming approach of [3] for hinging-hyperplane ARX systems, this could alleviate the limitations on the number of data and on the form of the model. Future work will also consider the framework of [7] in order to extend the algorithm to switched *nonlinear* regression, where dealing with large model structures becomes a critical issue.

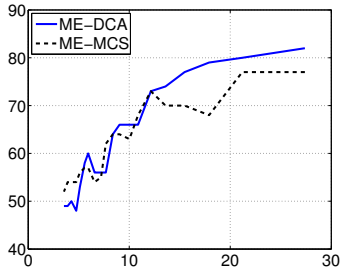

Fig. 1. Success rate (%) versus signal-to-noise ratio (dB).

REFERENCES

[1] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *IEEE CDC*, 2003, pp. 167–172.
[2] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.
[3] J. Roll, A. Bemporad, and L. Ljung, "Identification of piecewise affine systems via mixed-integer programming," *Automatica*, vol. 40, no. 1, pp. 37–50, 2004.
[4] A. L. Juloski, S. Weiland, and W. Heemels, "A Bayesian approach to identification of hybrid systems," *IEEE Trans. on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.
[5] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Trans. on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.
[6] F. Lauer, G. Bloch, and R. Vidal, "A continuous optimization framework for hybrid system identification," *Automatica*, vol. 47, no. 3, pp. 608–613, 2011.
[7] V. L. Le, G. Bloch, and F. Lauer, "Reduced-size kernel models for nonlinear hybrid system identification," *IEEE Trans. on Neural Networks*, vol. 22, no. 12, pp. 2398–2405, 2011.
[8] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668–677, 2011.
[9] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps, "A sparsification approach to set membership identification of switched affine systems," *IEEE Trans. on Automatic Control*, vol. 57, no. 3, pp. 634–648, 2012.
[10] F. Lauer, V. L. Le, and G. Bloch, "Learning smooth models of nonsmooth functions via convex optimization," in *Proc. of the IEEE Int. Workshop on Machine Learning for Signal Processing, Santander, Spain*, 2012.
[11] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.
[12] V. L. Le, F. Lauer, L. Bako, and G. Bloch, "Learning nonlinear hybrid systems: from sparse optimization to support vector regression," in *Proc. of the 16th ACM Int. Conf. on Hybrid Systems: Computation and Control, Philadelphia, PA, USA*, 2013, pp. 33–42.
[13] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search," *Journal of Global Optimization*, vol. 14, no. 4, pp. 331–355, 1999.
[14] H. A. Le Thi and T. Pham Dinh, "The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems," *Annals of Operations Research*, vol. 133, pp. 23–46, 2005.
[15] ——, "Solving a class of linearly constrained indefinite quadratic problems by DC algorithms," *Journal of Global Optimization*, vol. 11, no. 3, pp. 253–285, 1997.
[16] T. Pham Dinh and H. A. Le Thi, "Convex analysis approach to D.C. programming: Theory, algorithms and applications," *Acta Mathematica Vietnamica, dedicated to Professor Hoang Tuy on the occasion of his 70th birthday*, vol. 22, no. 1, pp. 289–355, 1997.
[17] ——, "DC optimization algorithm for solving the trust region subproblem," *SIAM Journal of Optimization*, vol. 8, no. 1, pp. 476–505, 1998.
[18] H. A. Le Thi, V. Huynh, and T. Pham Dinh, "Exact penalty and error bounds in DC programming," *Journal of Global Optimization*, vol. 52, no. 3, pp. 509–535, 2011.
[19] F. Lauer, "Estimating the probability of success of a simple algorithm for switched linear regression," *Nonlinear Analysis: Hybrid Systems*, vol. 8, pp. 31–47, 2013, supplementary material available at http://www.loria.fr/~lauer/klinreg/.